PharmaSUG 2022 - Paper DS-095

Making an ADaM Dataset Analysis-Ready

Sandra Minjoe, ICON PLC

ABSTRACT

One of the fundamental principles of ADaM is to be "analysis-ready". But what does that mean, and how do you determine if your analysis dataset is indeed "analysis-ready"?

This paper delves into what the ADaM documents say about being "analysis-ready", including what type of dataset manipulation is allowed (and not allowed) to happen between the ADaM dataset and the statistical output. It describes how to choose the appropriate dataset structure and recommends variables that will help efficiently create different types of analysis output, such as tables and figures. It also describes situations where "analysis-ready" doesn't apply. This paper includes examples of what you can do to ensure your dataset meets the ADaM "analysis-ready" fundamental principle.

INTRODUCTION

Study data flow of clinical trials is generally as shown in Figure 1:

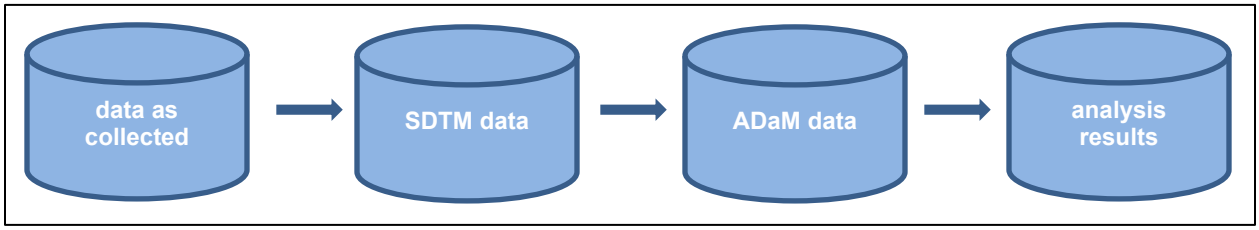


Figure 1: Study Data Flow

First, data as collected gets pushed into standard SDTM domains and variables, so that the same content can always be found in the same place. At the end, data needs to be reported in analysis results. The step between SDTM and analysis results is the ADaM data, which is the focus of this paper.

ADaM data is separated from analysis results generation for a number of reasons, including:

- **Efficiency**: An analysis dataset can be derived once and use for many different analyses.
- **Traceability**: ADaM data contains information to trace back to SDTM, which would be lost if the dataset wasn't saved.
- **Ease-of-Use**: An ADaM dataset can be consumed by someone with limited programming skills.

It's this last reason, ease-of-use, that is the highlight of the fundamental principle "analysis-ready".

WHAT DOES CDISC SAY?

The ADaM model document (v2.1)[1] Section 3.1, titled "Fundamental Principles" states the following:

> *Sponsors should strive to submit "analysis-ready" datasets, i.e., analysis datasets that have a structure and content that allows statistical analysis to be performed with minimal programming. An analysis-ready dataset is ready to be used directly by statistical analysis software with only minimal additional processing, for example a sorting of the observations or the selection of the appropriate records from the analysis dataset. No complex data manipulations such as transformations or transpositions are required to perform the supported analysis. This approach eliminates or greatly reduces the amount of programming required by analysts such as statistical reviewers.*

Later in the same section, it also states:

> *Note that within the context of ADaM, at a minimum analysis datasets contain the data needed for the review and re-creation of specific statistical analyses. It is not required that the data be collated into analysis-ready datasets solely to support data listings or other non-analytical displays, although some may choose to do so.*

ADaM Implementation Guide (IG) v1.3[1] Section 2.1 states:

> *ADaM datasets should have a structure and content that allow statistical analyses to be performed with minimal programming. Such datasets are described as "analysis-ready." ADaM datasets contain the data needed for the review and re-creation of specific statistical analyses. It is not necessary to collate data into analysis-ready datasets solely to support data listings or other non-analytical displays.*

The bottom line is that we need to ensure there is “minimal programming” needed to use ADaM datasets for creating specified statistical analyses.

MINIMAL PROGRAMMING

So, what constitutes “minimal programming”? From the CDISC documents described above, we know that sorting and subsetting are OK, but transformations and transpositions are not. Not mentioned, but certainly implied, is that merging datasets together is also not “minimal programming”.

Some refer to “minimal programming” as “one proc away”, which is a pretty good way to think about it. Within any statistical procedure that you would use to produce results, you can subset but you can’t transpose, merge, or create new variables.

Just remember that the “one proc away” concept applies to a specific number in the analysis results, not that every number on the table needs to be produced with a single procedure. Take, for example, a standard Adverse Event table, where you summarize overall events, events within body system, and events within preferred term, each across different treatment groups. Each categorization level probably needs its own procedure, plus you need another procedure to derive percentages, but the OCCDS dataset structure is considered analysis-ready for creating an Adverse Event table!

DATASET STRUCTURE

A big part of being analysis-ready has to do with dataset structure.

STANDARD ADAM STRUCTURES

Most of the time, standard structures ADSL, BDS, or OCCDS will be analysis-ready for a specific output:

- ADSL is designed to be analysis-ready for standard demography, baseline characteristics, and disposition tables.
- BDS is designed to be analysis-ready for standard change-from-baseline, shift, and time-to-event analyses. The ADaM Examples in Commonly Used Statistical Analysis Methods v1[1] also lists the following analysis methods that can be generated from BDS:
    - Analysis of Covariance (ANCOVA) and Analysis of Variance (ANOVA)
    - Chi-squared, including Chi-squared corrected
    - Cochran-Mantel-Haenszel and Mantel Haenszel
    - Fisher’s Exact
    - Kruskal-Wallis
    - Log Rank
    - McNemar
    - Mixed Models
    - Regression, including Cox, Linear, and Logistic
    - Sign Test
    - t-Test, both 1-sided and 2-sided

    - Wilcoxon (Mann-Whitney)
- OCCDS is designed to be analysis-ready for the typical occurrence analysis, where subjects within a hierarchy are counted. Standard adverse event, concomitant medications, medical history, and similar tables are all analysis-ready from OCCDS.

The standard structures ADSL, BDS, and OCCDS correspond to standard ADaM dataset classes SUBJECT-LEVEL ANALYSIS DATASET, BASIC DATA STRUCTURE, and OCCURRENCE DATA STRUCTURE, respectively.

CHOOSING AN ADAM STRUCTURE

In SDTM, the type of data collected (data topic or contents) determines the structure of the dataset. For example, adverse events and disposition are always in an SDTM Events structure, laboratory is always in an SDTM Findings structure, and Concomitant Medications and Exposure are always in an SDTM Interventions structure. However, an ADaM dataset structure is determined by how the data will be analyzed, not the data topic or contents. Table 1 shows the ADaM structure commonly used for different types of data:

Table 1: SDTM and Common ADaM Structures for Selected Data

SDTM Structure	Dataset Content	Common ADaM Structure
Events	AE data	Occurrence
	Disposition data	ADSL or BDS
Findings	Lab data	BDS
Interventions	Con Med data	Occurrence
	Exposure data	BDS

Notice that Table 1 shows the common ADaM dataset structures. Your structure may be different. Because the ADaM data structure chosen must be based on how it will be analyzed, in a specific study the structure shown here may not be appropriate! Do not decide on a dataset structure until you know what analysis needs to be produced.

NON-STANDARD ANALYSIS CLASS

While most analysis output can be easily derived from datasets in the structure of ADSL, BDS, or OCCDS, there are some situations where these structures don't fit the need. For example, multivariate Analysis of Variance which requires multiple independent variables on the same row, cannot be generated directly from the BDS structure. The ADaM Examples in Commonly Used Statistical Analysis

Methods v1[1] Section 2.6.3 states that a transpose, essentially putting multiple BDS parameters on a single row, would be required to make the data analysis-ready. Because this transposed dataset would not be of any of the standard ADaM structures, it would be of class "ADAM OTHER".

Unlike ADSL, BDS, and OCCDS described above, there is no corresponding structure for class ADAM OTHER. In the multivariate example described above, the structure would be more horizontal than BDS, but other datasets of class ADAM OTHER might need a different structure. Create your ADAM OTHER dataset in whatever structure would allow it to be analysis-ready for the output. Every ADaM dataset, even those of class ADAM OTHER, needs to follow all the fundamental principles of ADaM, including being analysis-ready.

VARIABLES TO AID IN ANALYSIS

Once you have decided on a dataset structure, you will then need to work on which variables to include in your dataset. Compared to SDTM, ADaM doesn't have many required variables, and instead many standard variables are permissible. This gives you a lot of flexibility!

As you design an ADaM dataset, consider how it will be used to create all of the analysis output, such as tables and figures. This mindset will enable you to create a dataset that makes table production as simple as possible, helping any reviewer (internal or external) better understand how the dataset was used to produce the output.

Some examples of what you can do to aid analysis include:

- Making text in your parameters look like the text needed on the analysis tables and figures.
- Creating timing variables that match the categories needed for analysis, and making the text of those variables look like the text needed on the analysis tables and figures.
- Including variables that allow you to arrange content on the tables and figures.
- Including variables that show which rows from the dataset are used to create specific analysis tables and figures.

Let's look at each of these topics in more detail.

PARAMETERS

Make text in your parameters look exactly like the text needed on the analysis tables and figures. When creating PARAM from SDTM data, you might need to change from the typically all-uppercase SDTM text into mixed case, plus tack on units, location, or other SDTM "qualifier" variables. Appropriately setting up the parameter in the dataset will avoid having to apply a format or doing any other work in the table or figure program to get the text as needed on the output. In addition to making the dataset easier to use for analysis, this can save a lot of time when producing multiple tables from the same dataset, since the parameter is defined once and used in multiple places, and reduces the places where typos could be made.

TIMING VARIABLES

Create timing variables that match the categories needed for analysis. Determine what every visit (or other timing variable) on the table represents, and use ADaM variables such as AVISIT to hold the text of this timing information. You might need to compare dates to assign analysis visits based on when the visit occurred (instead of the SDTM collected visit of when the visit should have occurred), and put unscheduled visits into visit windows. The description of what records will go into which analysis visit (or other timing variables) should be found in the Statistical Analysis Plan (SAP), and will need to be described in the variable derivation metadata and often in the Analysis Data Reviewer's Guide (ADRG).

Similar to the parameter description above, make the text in the timing variables look like the text needed on the analysis tables and figures. For example, even if you use collected SDTM visit information as the basis for the analysis visit, you probably need it to be in mixed case for the tables. Why not derive once,

in the dataset program, as it needs appear on the output, so that it doesn't have to be done in each output program? This will prevent having to apply a format, save time, and reduce typos.

VARIABLES USED FOR ARRANGING CONTENT ON THE OUTPUT

It's useful to include variables in the dataset that allow you to arrange content on the tables. For example, variables like RACEN, AVISITN, PARAMN, and AESEVN can be used to sort RACE, AVISIT, PARAM, and AESEV, respectively, in the order needed on the table.

Additionally, the variable AVAL can be used to sort the AVALC used for categorical analysis of scale variables, such as <Low, Medium, High> as shown in Table 2, or <Always, Often, Sometimes, Never>.

Table 2: Example: Creating AVAL Values to Sort AVALC

AVALC	AVAL
Low	1
Medium	2
High	3

You might be concerned about populating both AVAL and AVALC on the same row, however ADaMIG v1.3[1] Section 3.3.4.2 describes that creating a numeric AVAL to sort AVALC is an appropriate pairing, as long as the values of AVAL and AVALC are one-to-one.

In fact, numeric paired variables can be used for more than just sorting tables - they can also be useful in creating graphical displays. For this reason, you might want to consider assigning numbers to these variables that provide more information than ordinal values. Table 3 and Table 4**.** show examples of possible content in AVAL and AVISITN that would be useful for graphing data.

Table 3: Example: Creating AVAL Values to Sort and Graph AVALC

AVALC	AVAL
0-1	0.5
2-4	3
5-9	7
10-20	15
20-40	30

Table 4: Example: Creating AVISITN Values to Sort and Graph AVISIT

AVISIT	AVISITN
Baseline	0
Week 1	1
Week 4	4
Month 3	13
Month 6	26

In Table 3, notice that the ranges in AVALC are not the same size – specifically, the earlier ranges are smaller than the later ones. Trying to graph using an ordinal AVAL value of 1, 2, 3, 4, 5 would display as if all ranges were the same size, which is not appropriate. If you were to populate AVAL without considering

how this data would be used for graphing, there would need to be some code added to the graph program to derive something like what is shown for AVAL in Table 3. Rather than a simple ordinal value of 1, 2, 3, 4, 5, assigning AVAL to be the middle of each range of AVALC makes this dataset ready for graphing.

In Table 4, notice that the AVISIT values are not in the same units – here baseline doesn't have a unit, then there are some rows with units of weeks and others of months. Similar to the prior example, trying to graph using an ordinal value of 1, 2, 3, 4, 5 would display as if all visits were the same distance apart, which is not appropriate. If you were to populate AVISITN without considering how this data would be used for graphing, there would need to be some code added to the graph program to derive something like what is shown for AVISITN in Table 4. Rather than a simple ordinal value of 1, 2, 3, 4, 5, assigning the variable AVISITN to be the value in weeks represented by AVISIT makes this dataset ready for graphing.

In both Table 3 and Table 4, the numeric value allows the dataset to be analysis-ready for graphing. Also, the values of AVAL in Table 3 and the values of AVISITN in Table 4 can be used for ordering, so they are still analysis-ready for table analyses.

It helps to consider all the output needs, not just tables, before developing the ADaM dataset.

VARIABLES THAT CONNECT THE DATASET WITH THE OUTPUT

Flag variables ANLzzFL are used to show which rows from the ADaM dataset are used to create specific analysis table(s). You may have several different types of analyses coming from the same dataset, and variables ANL01FL, ANL02FL, etc. act as a form of analysis-results metadata, connecting specific rows in the dataset to the analysis table.

Table 5: Example: Using Analysis Flags ANLzzFL

row	AVISIT	AVISITN	AVAL	BASE	CHG	DTYPE	ANL01FL	ANL02FL
1	Baseline	0	114	114	0		Y	Y
2	Week 2	2	118	114	4		Y	Y
3	Week 2	2	126	114	12			
4	Week 4	4	122	114	8		Y	Y
5	Week 8	8	122	114	8	LOCF	Y	
6	Week 8	8	126	114	12	WOCF		Y
7	Week 12	12	134	114	20		Y	Y

Notice that in Table 5:

- Rows 2 and 3 are both windowed to AVISIT Week 2
  - Row 2 has ANL01FL = 'Y' and ANL02FL = 'Y', meaning it is the row used in both these analyses.
  - Row 3 has a missing value for ANL01FL and ANL02FL, meaning it is not used in either of these analyses.
- Rows 5 and 6 are both for AVISIT Week 8
  - Row 5 is used for Week 8 in any LOCF analysis as shown with ANL01FL="Y".
  - Row 6 is used for Week 8 in any WOCF analysis, as shown with ANL02FL="Y".
- Even though row 3 was not used for Week 2 analysis, it was the worst value prior to Week 8 so the value from that row is used for Week 8 WOCF analysis in row 6.

Labeling Variables

Did you know you can include text in the label of ANLzzFL variables, after the standard label text, to explain the analysis to be done using the flag? ADaMIG v1.3[1] describes this in Section 3.1.6. That means we might use the labels such shown in Table 6:

Table 6: Example ANLzzFL Labels

Variable Name	Variable Label
ANL01FL	Analysis Flag 01 – using LOCF
ANL02FL	Analysis Flag 02 – using WOCF

The inclusion of ANLzzFL variables, especially when those variables have descriptive labels, means it is very simple to choose the correct rows for specific analysis from the dataset. This makes the dataset analysis-ready for many different analyses.

LISTINGS AND OTHER NON-ANALYSIS NEEDS

The concept of "analysis-ready" applies only to ADaM datasets used for actual analysis. This means that datasets not used for analysis do not need to be "analysis-ready". Some common ADaM datasets that are not used for analysis are:

- Datasets used solely for listings
- Intermediate datasets that are not analyzed

Let's look at each of these separately.

LISTINGS

Because listings are not analysis, they do not need to be "analysis-ready". That means it is perfectly fine in listing program to transpose content from the ADaM dataset and to derive new variables. Also fine is to create a listing from an SDTM dataset merged with ADSL.

When determining whether an ADaM dataset would be useful for a listing, consider the following:

- **Does the listing support a specific analysis table?** If so, it would be useful to use the same dataset as was used in the table program.
- **Is the listing used to review collected data?** If so, it might be best to use unaltered SDTM data. This could be using
    1. an SDTM dataset + transpose of some SUPPQUAL content + some ADSL variables (e.g., MH + SUPPMH + ADSL), or
    2. an ADaM dataset that has copied over the SDTM variables unchanged (e.g., ADLB but using variables LBTEST, VISIT, LBDTC, and LBSTRESC)

It is not necessary to create an ADaM dataset when only listings are needed, but it is not "wrong" either. In fact, some companies require that all listings be created from ADaM.

If you have the option, evaluate whether it is worth the time and effort to create a "listing-ready" dataset. For example, a Drug Accountability ADaM dataset might be useful, even if no analysis tables are needed. That's because the SDTM DA dataset captures the amount dispensed on one row and amount returned on a different row, making it pretty complicated to subtract these values to determine amount taken. While this derivation could be done within the listing program, doing it in a dataset program would not only make the listing much easier to produce, it would also allow the dataset to undergo any standard company dataset validation.

In practice, you can often make your datasets "listing-ready", though listings that require concatenation of multiple variables into long text strings might be an exception. Since any submitted dataset must conform

to SAS transport file v5 requirements, there is a limit of 200 characters for all text variables. Only concatenate to larger than 200 characters in a saved dataset variable if the dataset will not be submitted (e.g., a dataset used only for the internal purpose of preparing for the listing).

INTERMEDIATE DATASETS

Some ADaM dataset are used solely as intermediate datasets and are not analyzed directly. Instead, the focus of these dataset is to split out some of the work when doing complex derivations, allowing for simpler explanations at each step. For datasets not used for analysis, there is no need to force the use of ADaM variables like AVAL, PARAM, and AVISIT.

ADaM intermediate datasets have been described in detail in other places, including CDISC Therapeutic Area User Guides (TAUGs), and United States Food and Drug Association (FDA) documents. Two types of intermediate datasets that we will discuss here are:

1. A dataset to collect dates, prior to a time-to-event ADaM dataset
2. A exposure dataset prior to an exposure summary dataset

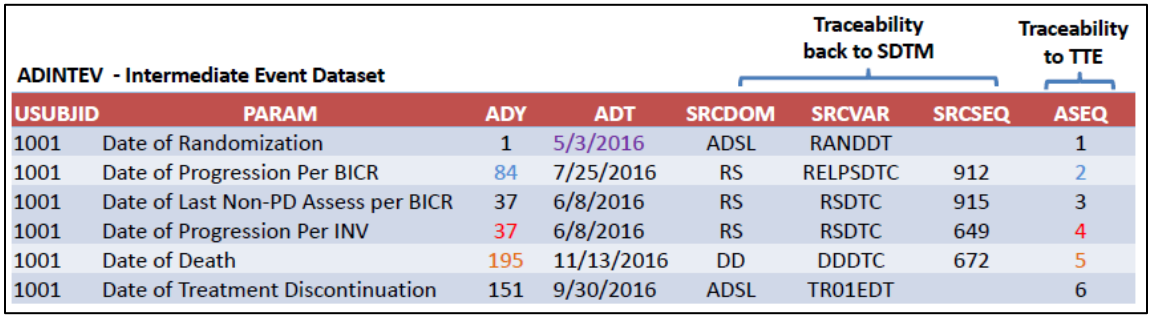
Let's review each of these examples and summarize their content and use.

Intermediate Dataset Examples: Collecting Dates Prior to Time-to-Event

It can be useful to create an intermediate dataset to collect dates, prior to a time-to-event ADaM dataset (often called ADTTE). When stored in a vertical structure, this intermediate dataset can be sorted by date and gives a snapshot of all the important events in the study. Examples of this type of intermediate dataset are in several places, including:

- **The Breast Cancer TAUG**[2] Section 5.3.2, published by CDISC in 2016. In this TAUG, the intermediate dataset ADDATES is a BDS structure and contains variables including:
    - ASEQ: sequential number used when referenced from dataset ADTTE
    - ASTDT: date the event occurred
    - PARAM: description of the event
    - PARAMCD: coded version of PARAM
    - AVALC: reported assessment associated with ASTDT
    - SRCDOM, SCRVAR, SRCSEQ: providing traceability back to datasets used as input to ADDATES
- **The Prostate Cancer TAUG**[3] Section 6.3.2, published by CDISC in 2017. This TAUG provides more current thinking of how to design an ADDATES datasets. Here ADDATES is of class ADAM OTHER (rather than BDS as in the Breast Cancer TAUG described above), since some required BDS variables aren't actually necessary for this intermediate dataset purpose. In this TAUG, ADDATES contains variables including:
    - ASEQ: sequential number used when referenced from dataset ADTTE *(same as from the Breast Cancer TAUG)*
    - ADT: date the event occurred *(rather than ASTDT from the Breast Cancer TAUG)*
    - ADTDESC: description of the analysis date *(rather than PARAM from the Breast Cancer TAUG)*
    - ADTDESCD: coded version of ADTDESC *(rather than PARAMCD from the Breast Cancer TAUG)*
    - SRCDOM, SCRVAR, SRCSEQ: providing traceability back to datasets used as input to ADDATES *(same as from the Breast Cancer TAUG)*

- **An FDA Presentation on Recommendations for Review-Ready Submissions to CDER**[4], given by Matilde Kam at the 2019 CDISC Interchange. On slide 21, it describes an intermediate dataset called ADINTEV, which looks to be BDS. It is similar in structure to the above-described TAUG ADDATES dataset, and contains variables ASEQ, ADT, PARAM, SRCDOM, SRCVAR, and SRCSEQ, as shown in Figure 2.



ADINTEV - Intermediate Event Dataset

USUBJID	PARAM	ADY	ADT	SRCDOM	SRCVAR	SRCSEQ	ASEQ
1001	Date of Randomization	1	5/3/2016	ADSL	RANDDT		1
1001	Date of Progression Per BICR	84	7/25/2016	RS	RELPSDTC	912	2
1001	Date of Last Non-PD Assess per BICR	37	6/8/2016	RS	RSDTC	915	3
1001	Date of Progression Per INV	37	6/8/2016	RS	RSDTC	649	4
1001	Date of Death	195	11/13/2016	DD	DDDTC	672	5
1001	Date of Treatment Discontinuation	151	9/30/2016	ADSL	TR01EDT		6

Figure 2: Dataset ADINTEV from FDA Presentation

In all three examples, the intermediate dataset prior to the time-to-event dataset has the same type of information in the same type of structure, though the variable names and dataset class differ.

Intermediate Dataset Example: Prior to Summary of Exposure

A exposure dataset (e.g., ADEX), prior to an exposure summary dataset (e.g. ADEXSUM), is described in the FDA Oncology Standard Safety Data Requests v1.3[5]. Here,

- Dataset ADEX looks to be a combination of SDTM variables from EX plus ADaM ADSL variables, plus ADaM numeric dates, plus a few requested FDA-specific derived variables. It appears to be one record per record found in SDTM dataset EX. Since it doesn't seem to be used for analysis, ADEX is probably best categorized as ADAM OTHER.
- Dataset ADEXSUM looks to be derived using ADEX as input. This dataset has variables PARAM, PARAMCD, AVAL, and AVALC, and contains parameters that would be used directly in analysis, so it appears to be BDS.

Intermediate Datasets Not Used Directly in Analysis

Intermediate datasets can be incredibly useful when creating a complex ADaM dataset. However, intermediate datasets are not often used directly in analysis themselves. Their purpose is to split out some of the derivations to make them easier to understand, provide traceability, and be useful for listings and review. Intermediate datasets not used in analysis tend to have very few variables: just the ones needed to help generate the content in the later ADaM dataset (e.g., ADTTE or ADEXPSUM), plus anything that will aid in review and traceability.

There is no need to make these types of intermediate datasets "analysis-ready". Instead, you can think of them as being "analysis-dataset-ready", since they are used to make it easier to derive the next ADaM dataset.

CONCLUSION

One of the requirements for any ADaM dataset is that it be analysis-ready. Choosing the appropriate dataset structure, as determined by the types of analyses that will be done with the data, is the first step.

Some other simple things you can do to help make your ADaM dataset analysis-ready include:

- **Making text variables, such as PARAM and character timing variables, look like what is needed on the table**. Not only does this avoid having to create formats, it also helps a reviewer understand what data was used to create the output, since the text in the dataset matches the text on the output.

- **Creating numeric variables to sort output**. Ordinal variables (1, 2, 3, 4, etc.) are often sufficient for ordering tables. Making the numbers representative of the data, rather than simply ordinal, can also help with graphing.
- **Including variables like ANLzzFL that filter to the rows used in the output**. This helps the output programmer choose the correct rows, and it helps a reviewer see the connection between the dataset and the output.

The concept of "analysis-ready" only applies to ADaM datasets that will be used directly for analysis. ADaM datasets uses solely for listings or as intermediate datasets do not need to be analysis-ready.

REFERENCES

1. CDISC. "ADaM." Accessed March 11, 2022. https://www.cdisc.org/standards/foundational/adam. *All CDISC ADaM documents (model, IG, Statistical Examples document, etc.) can be accessed from this site.*
2. CDISC. "Breast Cancer Therapeutic Area User Guide v1.0". Published 2016. Accessed March 11, 2022. https://www.cdisc.org/standards/therapeutic-areas/breast-cancer.
3. CDISC. "Prostate Cancer Therapeutic Area User Guide v1.0". Published 2017. Accessed March 11, 2022. https://www.cdisc.org/standards/therapeutic-areas/prostate-cancer.
4. Kam, Matilde. 2019. "FDA Review Process: Recommendations for Review-Ready Submissions to CDER". CDISC Annual Interchange 2019 Proceedings. Available to CDISC members at https://www.cdisc.org/system/files/all/event/restricted/2019_US/Session_8/MKamCDISCUSInt2019_04OCT2019_FINAL.pdf.
5. FDA. "Pilot OCE/OOD Standard Safety Data Requests v1.3". Published 2021. Accessed March 11, 2022. https://www.fda.gov/media/133252/download.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sandra Minjoe
MinjoeSandra@prahs.com